

Listing of Claims:

1. (previously presented) A system for providing multiple language support for at least one application program, the system comprising:

a plurality of language resource bundles comprising associations between language keys and displayable language-sensitive elements, each resource bundle corresponding to a different language, wherein at least one association is specific to a particular application program and at least one association is applicable to a plurality of different application programs; and

a language resource manager configured to receive a first language key from a first application program, locate a language resource bundle corresponding to a currently-selected language, identify a language-sensitive element associated with the first language key and the first application program, and provide the identified language-sensitive element to the first application program for display in a graphical user interface.

2. (previously presented) The system of claim 1, wherein the first application program is configured to provide a language key to the language resource manager, receive a language-sensitive element from the language resource manager, and display the language-sensitive element in a graphical user interface.

Best Available Copy

3. (original) The system of claim 1, wherein at least one language-sensitive element is selected from the group consisting of a text string, an icon, a graphic, and a video clip.

4. (previously presented) The system of claim 1, wherein the language resource manager is further configured to display a language switching mechanism in the graphical user interface for changing the currently-selected language in response to user input.

5. (previously presented) The system of claim 4, wherein the language switching mechanism is selected from the group consisting of a drop-down list, a menu, a button, an edit box, and an icon.

6. (previously presented) The system of claim 1, wherein the language resource manager is further configured to change the currently-selected language in response to at least one keystroke.

7. (original) The system of claim 1, further comprising:
a language switching component configured, in response to a change in the currently-selected language, to send to the language resource manager a language key corresponding to a first language-sensitive element displayed in the graphical user interface, receive from the language resource manager a second language-

Best Available Copy

sensitive element, and replace the first language-sensitive element with the second language-sensitive element in the graphical user interface.

8. (original) The system of claim 7, wherein the language switching component is further configured to replace each language-sensitive element displayed in the graphical user interface with a new language-sensitive element in response to a change in the currently-selected language.

9. (previously presented) The system of claim 7, wherein the language switching component is further configured to preempt the first application program, save a state of the first application program, discard the graphical user interface being currently displayed, generate a new graphical user interface comprising at least one new language-sensitive element provided by the language resource manager, restore the state of the first application program, and resume execution of the first application program.

10. (original) The system of claim 1, wherein the language resource manager is in communication with a plurality of applications to receive language keys and provide language-sensitive elements.

11-12. (cancelled).

Best Available Copy

13. (original) The system of claim 1, further comprising:
a parser configured to parse a language resource file comprising descriptors of language keys and descriptors of language-sensitive elements and to generate therefrom a language resource bundle.
14. (original) The system of claim 13, wherein the language resource file comprises human-readable text.
15. (original) The system of claim 13, wherein at least one descriptor of a language key is selected from the group consisting of a string, a character, a number, and a symbol.
16. (original) The system of claim 13, wherein at least one descriptor of a language-sensitive element comprises a Unicode string.
17. (original) The system of claim 13, wherein at least one descriptor of a language-sensitive element comprises an address.
18. (original) The system of claim 17, wherein the address comprises a file name.
19. (original) The system of claim 17, wherein the address comprises a uniform resource locator (URL).

20. (previously presented) The system of claim 1, wherein the language resource manager is a component of a framework used by a plurality of application programs.

21. (previously presented) A method for providing multiple language support for at least one application program, the method comprising:

providing a plurality of language resource bundles comprising associations between language keys and displayable language-sensitive elements, each resource bundle corresponding to a different language, wherein at least one association is specific to a particular application program and at least one association is applicable to a plurality of different application programs;

receiving a first language key from a first application program,

locating a language resource bundle corresponding to a currently-selected language;

identifying a language-sensitive element associated with the first language key and the first application program; and

providing the identified language-sensitive element to the first application program for display in a graphical user interface.

22. (original) The method of claim 21, further comprising:

displaying the language-sensitive element in a graphical user interface.

23. (original) The method of claim 21, wherein at least one language-sensitive element is selected from the group consisting of a text string, an icon, a graphic, and a video clip.

24. (original) The method of claim 21, further comprising:
displaying the language switching mechanism in the graphical user interface for changing the currently-selected language in response to user input.

25. (original) The method of claim 24, wherein the language switching mechanism is selected from the group consisting of a drop-down list, a menu, a button, an edit box, and an icon.

26. (original) The method of claim 21, further comprising:
changing the currently-selected language in response to at least one keystroke.

27. (original) The method of claim 21, further comprising:
in response to a change in the currently-selected language:
 sending a language key corresponding to a first language-sensitive element displayed in the graphical user interface;
 receiving a second language-sensitive element in response to the language key; and

replacing the first language-sensitive element with the second language-sensitive element in the graphical user interface.

28. (original) The method of claim 27, further comprising:

replacing each language-sensitive displayed in the graphical user with a new language-sensitive element in response to a change in the currently-selected language.

29. (previously presented) The method of claim 27, further comprising:

preempting the first application program;

saving a state of the first application program;

discarding the graphical user interface being currently displayed;

generating a new graphical user interface comprising at least one new language-sensitive element received in response to the language key;

restoring the state of the first application program; and

resuming execution of the first application program.

30. (previously presented) The method of claim 21, further comprising:

receiving language keys from a plurality of application programs; and

providing corresponding language-sensitive elements to each application program.

31-32. (cancelled).

33. (original) The method of claim 21, further comprising:
parsing a language resource file comprising descriptors of language keys and
descriptors of language-sensitive elements to generate therefrom a language
resource bundle.
34. (original) The method of claim 33, wherein the language resource file
comprises human-readable text.
35. (original) The method of claim 33, wherein at least one descriptor of a
language key is selected from the group consisting of a string, a character, a number,
and a symbol.
36. (original) The method of claim 33, wherein at least one descriptor of a
language-sensitive element comprises a Unicode string.
37. (original) The method of claim 33, wherein at least one descriptor of a
language-sensitive element comprises an address.
38. (original) The method of claim 37, wherein the address comprises a file
name.

39. (original) The method of claim 37, wherein the address comprises a uniform resource locator (URL).

40. (previously presented) A computer program product for providing multiple language support for at least one application program, the computer program product comprising:

a plurality of language bundles, each language bundle corresponding to a particular language and comprising associations between language keys and displayable language-sensitive elements, wherein at least one association is specific to a particular application program and at least one association is applicable to a plurality of different application programs;

program code for receiving a first language key from a first application program;

program code for locating a language resource bundle corresponding to a currently-selected language;

program code identifying a language-sensitive element associated with the first language key and the first application program; and

program code for providing the identified language-sensitive element to the first application program for display in a graphical user interface.

41-59. (cancelled).

60. (previously presented) A system for providing multiple language support for at least one application program, the system comprising:

a plurality of language resource bundles comprising associations between language keys and displayable language-sensitive elements, each resource bundle corresponding to a different language;

a language switching component to preempt an application program, save a state of the application program, discard the graphical user interface being currently displayed, generate a new graphical user interface comprising at least one new language-sensitive element indicated by a language resource bundle for a received language key, restore the state of the application program, and resume execution of the application program.

61. (previously presented) The system of claim 60, further comprising:

a parser configured to parse a language resource file comprising descriptors of language keys and descriptors of language-sensitive elements and to generate therefrom a language resource bundle.

62. (previously presented) The system of claim 61, wherein the language resource file comprises human-readable text.

63. (previously presented) A method for providing multiple language support for at least one application program, the method comprising:

providing a plurality of language resource bundles comprising associations between language keys and displayable language-sensitive elements, each resource bundle corresponding to a different language;

receiving a first language key;

locating a language resource bundle corresponding to a currently-selected language;

identifying a language-sensitive element associated with the first language key;

preempting the application program;

saving a state of the application program;

discarding the graphical user interface being currently displayed by the application program;

generating a new graphical user interface for the application program comprising at least one new language-sensitive element indicated by the located language resource bundle for the first language key;

restoring the state of the application program; and

resuming execution of the application program.

64. (previously presented) The method of claim 63, further comprising:

parsing a language resource file comprising descriptors of language keys and descriptors of language-sensitive elements and to generate therefrom a language resource bundle.

65. (previously presented) The method of claim 64, wherein the language resource file comprises human-readable text.

66. (previously presented) A system for providing multiple language support for at least one application program, the system comprising:

a parser to parse a language resource file written in human-readable text and comprising descriptors of language keys and descriptors of language-sensitive elements and generate therefrom a language resource bundle comprising associations between language keys and displayable language-sensitive elements for a particular language; wherein a language switching component is to preempt the application program, save a state of the application program, discard the graphical user interface being currently displayed, generate a new graphical user interface comprising at least one new language-sensitive element indicated by a corresponding language resource bundle for a received language key, restore the state of the application program, and resume execution of the application program.

67. (previously presented) The system of claim 66, wherein at least one descriptor of a language key is selected from the group consisting of a string, a character, a number, and a symbol.

68. (previously presented) The system of claim 66, wherein at least one descriptor of a language-sensitive element comprises a Unicode string.

69. (previously presented) The system of claim 66, wherein at least one descriptor of a language-sensitive element comprises an address.

70. (previously presented) The system of claim 69, wherein the address comprises a file name.

71. (previously presented) The system of claim 69, wherein the address comprises a uniform resource locator (URL).

Best Available Copy